REMARKS

Claims 1, 3-5, 7-12 and 15-33 are pending, of which claims 15 and 21 are independent method claims with corresponding computer program product claims 27 and 28. As indicated above, claims 1, 7, 11, 15, 20, 21, 28, and 33 have been amended and claims 2, 6, and 13-14 have been canceled without prejudice by this paper. Applicants note for the record that the subject matter of claims 2, 6, and 13-14 is covered by other pending claims, and therefore canceling these claims does not evince an intent to surrender any subject matter. Applicants numbering of the claims in this paper is consistent with the numbering of claims in the Office Action as renumbered due to Applicants having skipped the number "13" in the claims originally filed.

The Office Action rejected independent claims 1, 15 and 28 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,272,492 to Kay ("*Kay*") and rejected independent claims 21 and 27 under 35 U.S.C. § 103(a) as being unpatentable over *Kay* in view of U.S. Patent No. 6,339,423 to Sampson et al. ("*Sampson*"). The dependent claims were rejected under 35 U.S.C. § 102(e) as being anticipated by *Kay* or under 35 U.S.C. § 103(a) as being unpatentable over *Kay* in view of *Sampson*.[1]

Applicants' invention, as claimed for example in independent method claim 15, relates to directing a client request to a particular back end server. The method identifies the particular back end server from a directory if the client request is for a private or home public folder that only the particular back end server stores. The method identifies the particular back end server from a list of a plurality of back end servers capable of servicing the client request, using a selection module, if the client request is for an application public folder. The selection module identifies the same particular back end server from the list for each client request of a particular user. The method also directs the client request to the particular back end server.

Applicants invention, as claimed for example in independent method claim 21, also relates to directing a client request for content in an application public folder to a particular back end server. The method identifies a security token of a user associated with the client request and identifies a list of back end servers from a directory—each back end server in the list of back

[1] Although the prior art status of the cited art is not being challenged at this time, Applicants reserve the right to do so in the future. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status or asserted teachings of the cited art.

end servers has a replica of the application public folder. The method selects the particular back end server from the list using a selection module that hashes the security token of the user to identify the particular back end server within the list of back end servers. The method directs the client request to the particular back end server selected by the selection module.

Kay discloses a front-end proxy server that receives a content request from a user system, processes the content request, and transmits the processed content request to a back-end server. Col. 1, ll. 43-46; Figure 1a. The front-end proxy server receives a content item corresponding to the processed content request from the back-end server and transmits the content item to the user system. Col. 1, ll. 46-55; Figure 1a. A mapping table within the front-end proxy server includes fields to map a received request to one or more back-end servers. Col. 4, ll. 30-39; Figure 1b & 1d. As described in connection with Figures 5a and 5b, *Kay's* front-end proxy server also may be used to implement transparent extended searching capabilities. Col. 6, l. 14- col. 7, l. 8.

The Office Action asserts that search engine 508 of Figure 5a in *Kay* discloses a selection module that identifies a back end server. *See, e.g.,* Office Action p. 5 (rejection of claim 15). Applicants note however, that *Kay*'s search engine is not used to identify a back-end server. Rather, as described in connection with Figures 5a and 5b, front-end proxy server 502 receives a request for a content object, such as a document, graphic, program, etc., from a web browser application program operated by a user. Col. 6, ll. 36-41. Search proxy server 502 processes the request, transmits the processed request to one or more back-end servers, which handles the request by retrieving the content object specified by the request, and transmits the retrieved content to the user. Col. 6, ll. 41-47. The search proxy server 502 also performs topical analysis on the content items received from the back-end servers, generates one or more search requests based on the extracted topical content, transmits the search requests to search engine 508 which performs the searches on other web servers 516, receives the responses from the search engine 508, and transmits all or part of the search responses to the user. Col. 6, l. 50 - col. 7, l. 7. The searching is transparent because the user did not have to make an explicit search request. Nevertheless, as described above, *Kay's* search engine is not used to identify a back end server. Accordingly, among other things, *Kay* fails to teach, suggest, or motivate identifying a particular back end server, from a list of a plurality of back end servers capable of servicing a client request, using a selection module, if the client request is for an application public folder, wherein the selection module identifies the same particular back end server from the list for each client

request of a particular user. In fact, *Kay* provides no support whatsoever for making any distinctions between private, home public, and application folders. Accordingly, Applicants respectfully submit that the rejection of independent method claim 15 and corresponding computer program product claim 28 is improper and should be withdrawn.

Sampson teaches a single access control system to manage access by users to resources that belong to multiple domains. Col. 3, ll. 20-22. In *Sampson*, a server in one domain transmits a data token to a client seeking access to a resource in another domain. Col. 3, ll. 25-26. The client transmits the data token to a server in the other domain, which uses the data token to verify that the user is authorized to access resources protected by the control system, and then transmits access control cookies to the client. Col. 3, ll. 27-32.

"To establish a *prima facie* case of obviousness . . . the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP § 706.02(j). The Office Action asserts that column 8, lines 15-30 of *Sampson* disclose a selection module comprising a hash function for hashing a security token to select a particular back end server from a list. *See* Office Action, p. 8 (rejection of claim 6) and p. 11 (rejection of claim 21). The Office Action's reasoning to justify this conclusion begins by simply stating that "[t]he process of using a hash and a security token to select a server is well known in the art." Office Action, p. 8 (rejection of claim 6). Applicants respectfully disagree. As support, the Office Action notes that column 8, lines 15-30 of *Sampson* teach multi-domain access control in which a hash and a security token are used to access different domains. *Id.* The Office Action's logic continues with "[i]t would have been obvious to some one of ordinary skill at the time of the invention to use a hash and security token to insure that only authorized users are allowed to access secured information." *Id.*

While perhaps true, using a hash and security token to insure that only authorized users are allowed to access secured information is <u>not</u> even the claim limitation at issue and is quite different from hashing a security token to select a back end server from a list of back end servers in the first instance. For example, independent claim 21 recites selecting a particular back end server from a list using a selection module that hashes the security token of a user to identify the particular back end server within the list of back end servers. As a result, the Office Action fails to even allege that *Sampson* teaches a selection module that hashes the security token of a user to identify a particular back end server from a list of back end servers. Furthermore, *Sampson's* use

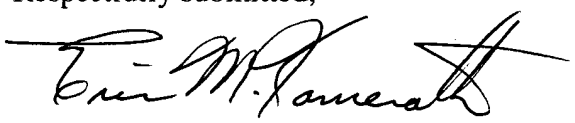
of a token and hash for access control provides no support whatsoever for hashing a security token to select a back end server from a list of back end servers. Among other things, therefore, *Sampson* fails to teach, suggest, or motivate selecting a particular back end server from a list using a selection module that hashes the security token of a user to identify the particular back end server within the list of back end servers. Accordingly, Applicants respectfully submit that the rejection of independent method claim 21 and corresponding computer program product claim 27 is improper and should be withdrawn.

Based on at least the foregoing reasons, therefore, Applicants respectfully submit that the cited prior art fails to anticipate or make obvious Applicants invention, as claimed for example, in independent claims 15, 21, 27, and 28. Applicants note for the record that the remarks above render the remaining rejections of record for the independent and dependent claims moot, and thus addressing individual rejections or assertion with respect to the teachings of the cited art is unnecessary at the present time, but may be undertaken in the future if necessary or desirable, and Applicants reserve the right to do so.

In the event that the Examiner finds any remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 12th day of April, 2004.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
ERIC M. KAMERATH
Registration No. 46,081
Attorneys for Applicant

Customer No. 022913

RDN:EMK:cm
KCS0000000745V001